Jurnal Komputer dan Informatika
Vol 20 No 1, April 2025: hlm 49 - 61

ISSN 1410-7228 (Versi Cetak)
ISSN 2716-4233 (Versi Elektronik)

PERBANDINGAN ALGORITMA SVM LINEAR, KNN, DAN DECISION TREE DALAM MENGKLASIFIKASI KUALITAS AIR

Gabriel Nathanael Irawan
Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara,
Jln. Letjen S. Parman No. 1, Jakarta, 11440, Indonesia
E-mail: gabriel.535220142@stu.untar.ac.id,

ABSTRAK

Air merupakan kebutuhan utama manusia. Banyak dari masyarakat yang mengandalkan sungai maupun air tanah sebagai sumber air. Namun, sebagian besar air yang ada sudah tercemar oleh limbah domestik maupun limbah industri. Penting untuk memperhatikan kualitas air untuk konsumsi karena, akibat dari mengkonsumsi air yang tercemar sendiri mulai dari diare, kolera, disentri, tipoid, dan sebagainya. Tujuan penelitian ini adalah untuk membandingkan 3 algoritma yaitu SVM *Linear*, KNN, dan *decision tree* dalam performanya mengklasifikasi dataset *Water Quality*, dengan berbagai skenario. Hasil eksperimen menunjukan bahwa *decision tree* memperoleh hasil paling baik dalam kasus ini dengan nilai rata-rata akurasi 0,6387 pada komposisi data latih 80%. Sedangkan algoritma SVM *Linear* merupakan algoritma paling buruk dengan nilai rata-rata akurasi 0,58625 pada komposisi data latih 60%.

Kata kunci : Klasifikasi, Kualitas Air, SVM *Linear*, KNN, Pohon Keputusan

ABSTRACT

Water is a basic necessity for humans. Many rely on rivers and groundwater as water sources. However, most of the water available is already contaminated by domestic and industrial waste. It is important to pay attention to water quality for consumption because consuming polluted water can lead to diseases such as diarrhea, cholera, dysentery, typhoid, and so on. The purpose of this research is to compare the performance of three algorithms: Linear SVM, KNN, and decision tree in classifying the Water Quality dataset under various scenarios. The experimental results show that the decision tree performs the best in this case with an average accuracy of 0.6387 on a 80% training data composition. Meanwhile, the Linear SVM algorithm performs the worst with an average accuracy of 0.58625 on a 60% training data composition.

Keywords*: Classificatoin, Water Quality, SVM Linear, KNN, Decision Tree*

1. PENDAHULUAN

Air bersih sangat penting bagi kehidupan manusia, termasuk dalam keperluan industri dan pasokan air PDAM (Perusahaan Daerah Air Minum). Di beberapa daerah, sungai menjadi sumber utama untuk mendapatkan air bersih. Namun, masalah pencemaran yang disebabkan oleh limbah domestik dan limbah industri telah mengakibatkan penurunan kualitas air di beberapa titik sungai. Tidak hanya berdampak pada lingkungan, pencemaran air ini juga berpotensi berdampak pada kesehatan individu dan Masyarakat, terutama di negara seperti Indonesia [1].Selain berasal dari sungai, sumber air bersih dapat berasal dari air tanah, baik air tanah dalam maupun air tanah dangkal. Air sendiri meskipun terlihat jernih dan tidak berbau atau berasa, belum menjamin keamanan air untuk dikonsumsi [2].

Penting untuk memastikan bahwa air yang dikonsumsi benar-benar memiliki kualitas yang baik dan aman karena, manusia sendiri memanfaatkan air untuk kehidupan sehari-hari seperti

memasak, mencuci, dan sebagai sumber hidrasi. Karena selain dapat menimbulkan sumber penyakit, air juga merupakan media penyebaran penyakit yang sangat luas [3]. Di era teknologi yang makin berkembang ini juga turut membuat air tercemar sehingga memiliki kualitas yang buruk. Walaupun memiliki standar yang berbeda di setiap negara, tetapi menurut WHO setidaknya perlu pasokan air sebanyak 30-60 liter per hari untuk negara berkembang khususnya seperti negara Indonesia [4]. Walaupun sebagai besar daerah Indonesia secara geografis terdiri dari air, masih banyak daerah di Indonesia yang kekurangan air bersih [5]. Kualitas air cenderung berubah, apalagi air sungai yang membawa hasil limbah aktivitas domestik dan industri [6].

Sungai yang menjadi sumber mendapatkan air bersih bagi sebagian masyarakat menjadi tercemar akibat pembuangan limbah dari berbagai aktivitas domestik, seperti mandi dan mencuci. Selain itu limbah industri juga kerap membuang limbahnya ke sungai yang membuat tekanan terhadap ekosisteme air meningkat yang membuat keberlangsungan hidup lingkungan perairan menjadi terancam [7]. Pertumbuhan penduduk juga berpengaruh terhadap kualitas air karena dengan bertambahnya jumlah penduduk, maka tinggi juga aktivitas manusia yang dapat menyebabkan pencemaran lingkungan [8]. Kebutuhan akan air bersih tidak hanya bagi kegiatan rumah tangga, di sektor lain seperti pertanian, pariwisata, industri, dan pertambangan juga membutuhkan kualitas air yang baik. Air yang memiliki kualitas buruk dapat menyebabkan penyakit seperti diare, kolera, disentri, tipoid, dan sebagainya [9]. Pencemaran air sendiri dapat diartikan sebagai turunnya kualitas air sampai ke tingkat di mana air tersebut tidak dapat digunakan sebagai semestinya [10].

Dampak dari pencemaran air tersebut memiliki konsekuensi bagi kualitas air yang tersedia untuk dikonsumsi. Oleh karena itu penelitian ini bertujuan untuk membandingkan 3 algoritma yaitu *Support Vector Machine* (SVM), *K-Nearest Neighbors* (KNN), dan *Decision Tree*, dalam mengklasifikasikan air yang aman untuk diminum dan air yang tidak aman diminum, berdasarkan data *Water Quality*. Tujuan dari penelitian ini adalah untuk menemukan algoritma yang paling baik berdasarkan nilai akurasinya. Hasil penelitian ini diharapkan dapat bermanfaat dalam memberikan informasi mengenai algoritma yang paling cocok digunakan pada kasus data *Water Quality*.

2. METODE PENELITIAN

2.1 Diagram Alir

Gambar 1 Diagram Alir Penelitian

Tahap pertama dalam melakukan penelitian ini adalah identifikasi masalah. Identifikasi masalah pada penelitian ini terkait dengan penjelasan air sebagai kebutuhan utama manusia, namun air sendiri mulai tercemar, sehingga terdapat air yang tidak aman untuk diminum dan air yang aman untuk diminum. Tahap kedua adalah melakukan studi literatur. Pada tahap ini mulai melakukan pencarian mengenai informasi yang berkaitan dengan penelitian ini dalam bentuk jurnal. Tahap ketiga adalah pengumpulan data. Data diperoleh melalui link https://www.kaggle.com/datasets/adityakadiwal/*water-potability*/data. Data tersebut merupakan data kualitas air yang terdiri dari 9 fitur dan 1 label.

Tahap keempat adalah pengolahan data. Pada tahap ini, data diolah terlebih dahulu sebelum melakukan klasifikasi dengan algortima. Tahap kelima adalah hasil dan pembahasan. Pada tahap ini,

ISSN 1410-7228 (Versi Cetak)
ISSN 2716-4233 (Versi Elektronik)

data sudah diproses dengan algortima, lalu ditampilkan hasil perbandingan dari 3 algoritma yang digunakan. Tahap keenam atau yang terakhir adalah kesimpulan dan saran. Pada bagian ini, algoritma yang paling baik dalam mengklasifikasi kualitas air dengan data *Water Quality* akan ditampilkan dan juga memberikan saran untuk pengembangan selanjutnya.

2.2 Algoritma *Support Vector Machine* (SVM)

Support Vector Machine (SVM) adalah salah satu teknik klasifikasi yang cara kerjanya dengan melakukan pemetaan data ke dimensi yang lebih tinggi menggunakan teknik kernel [11]. Salah satu kekurangan algoritma SVM adalah lamanya waktu yang dibutuhkan untuk pelatihan model. Selain pelatihan model yang memakan waktu yang lama, mencari nilai *hyperparameter* yang mengoptimalkan nilai akurasi juga merupakan salah satu tantangannya. Karena waktu pelatihan yang lama, pencarian nilai *hyperparameter* juga menjadi proses yang memakan waktu [12]. Pencarian nilai *hyperparameter* yang paling baik dapat menggunakan teknik *grid search CV*. Metode ini mengkombinasikan *hyperparameter* dan validasi untuk setiap kombinasi [13]. *Grid search CV* akan melakukan pengujian terhadap beberapa nilai *hyperparameter* yang sudah diinisialisasikan, kemudian akan membandingkan *hyperparameter* tersebut,mana yang memberikan nilai akurasi terbaik. Penelitian ini menggunakan SVM dengan kernel *linear*. Nilai *hyperparameter* yang diuji coba menggunakan metode *grid search* CV untuk dilihat yang paling baik berdasarkan nilai akurasi adalah 0.001, 0.01, 0.1, 1, 10, 100.

Persamaan (1) adalah rumus untuk mencari nilai α, di mana $\emptyset(\vec{\alpha})$ adalah nilai α yang dicari, kemudian $\sum_{i=1}^{N} \alpha_i$ adalah jumlah nilai α yang dimulai dari α_i sampai α_N, lalu terdapat $\sum_{i=1,j=1}^{N} \alpha_i \alpha_j \, y_j y_j (\vec{x}_i \cdot \vec{x}_j)$, di mana $\vec{x}_i$ dan $\vec{x}_j$ adalah nilai *vector* suatu matriks, dan y sendiri adalah label dari masing masing matriks.

$$\emptyset(\vec{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{N} \alpha_i \alpha_j \, y_j y_j (\vec{x}_i \cdot \vec{x}_j) \tag{1}$$

Persamaan (2) merupakan rumus untuk menghitung vektor *weight*. Nilai α, adalah nilai alfa yang dapat dicari menggunakan persamaan (1). Kemudian akan dilakukan perkalian antara α, y, dan $\vec{x}$, sejumlah nilai *N* yang ada.

$$\vec{\omega} = \sum_{i=1}^{N} \alpha_i y_i \vec{x}_i \tag{2}$$

Persamaan (3) merupakan rumus untuk menghitung bias. Nilai $\vec{\omega}$ merupakan nilai vektor *weight* yang dapat dicari menggunakan persamaan (2). Nilai $\vec{x}_i$ adalah nilai vektor suatu matriks. $\vec{\omega}$ akan menggambil nilai minuman dan maksimumnya.

$$b = \frac{1}{2} \left(\min_{i:y_{i=+1}} (\vec{\omega} \cdot \vec{x}_i) + \max_{i:y_{i=-1}} (\vec{\omega} \cdot \vec{x}_i) \right) \tag{3}$$

Persamaan (4) merupakan fungsi SVM *classifier*. Nilai $\vec{\omega}$ merupakan vektor *weight* yang dicari dengan persamaan (2), nilai $\vec{x}$ merupakan vektor, dan *b* adalah nilai bias yang dicari dengan persamaan (3).

$$f(\vec{x}) = \vec{\omega} \cdot \vec{x} - b \tag{4}$$

2.3 Algoritma *K-Nearest Neighbors* (KNN)

K-Nearest Neighbors (KNN) adalah suatu model pembelajaran *supervised learning* yang digunakan untuk klasifikasi data. KNN bekerja dengan menemukan objek terdekat dalam data pelatihan dengan objek baru yang diuji, berdasarkan jarak *euclidean* [14]. Jarak *Euclidean* sendiri

merupakan sebuah rumus yang membandingkan jarak antara 2 titik. Dalam kasus ini kedua titik tersebut adalah data latih dan data uji [15]. Tahapan awal dalam menggunakan metode KNN adalah menentukan nilai *k* yaitu jumlah tetangga terdekat yang digunakan untuk klasifikasi [16]. Nilai *k* yang paling baik berbeda tergantung datanya. Pengguanaan metode *grid search CV* juga digunakan dalam pencarian nilai *k* untuk klasifikasi KNN dengan melihat nilai akurasinya. Nilai *k* yang dipakai dalam metode *grid search CV* untuk melihat mana yang paling baik berdasarkan nilai akurasinya adalah rentang angka dari 1 sampai 21.

Persamaan (5) merupakan rumus untuk menghitung jarak *euclidean*. Nilai *d* merupakan hasil perhitungan jaraknya, nilai *x* dan *y* adalah nilai sebuah titik.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{5}$$

2.4 Algoritma *Decision Tree*

Decision Tree merupakan suatu model algoritma yang berbentuk seperti pohon yang terdiri dari akar (*root*), cabang, dan daun (*leaf node*). Akar sendiri merupakan titik awal dari pohon yang menggambarkan keseluruhan permasalahan yang ingin diselesaikan. Cabang pada *decision tree* merupakan pilihan yang ada, daun adalah keputusan yang dipilih [17]. *Decision tree* menggambarkan keputusan yang ada lalu mengatur setiap kondisi yang ada. Struktur *decision tree* sendiri seperti diagram alur yang berbentuk pohon di mana cara kerjanya dimulai dari akar, lalu bergerak sampai ke daun yang merupakan keputusan akhir [18]. Dalam algoritma *decision tree*, *entropy* dan *information gain* berperang penting, di mana *entropy* mengukur tingkat keberagaman data, sedangkan information gain merupakan parameter efektivitas yang digunakan untuk mengevaluasi seberapa baik sebuah atribut dapat mengklasifikasikan data [19].

Persamaan (6) merupakan rumus mencari nilai *entropy*, di mana *S* adalah himpunan dari data, c adalah banyaknya partisi (jumlah kelas) *S* dan $p_i$adalah probabilitas solusi positif maupun negatif.

$$E(S) = \sum_{i>1}^{c} -p_i \log_2 p_i \tag{6}$$

Persamaan (7) adalah rumus untuk mencari nilai *information gain*, di mana *A* adalah atribut, *S* merupakan himpunan data, *n* jumlah partisi dari *A*, $|S_i|$ merupakan jumlah sampel untuk nilai *i*, $|S|$ merupakan jumlah seluruh sampel data, dan *E(S)* merupakan nilai *entropy* yang dihitung melalui persamaan (6).

$$Gain(S, A) = E(S) - \sum_{i=1}^{n} \left|\frac{Si}{S}\right| E(S_i) \tag{7}$$

2.5 *Confusion Matrix*

Confusion matrix merupakan suatu metode evaluasi yang memvisualisasikan performa model klasifikasi, dengan membandingkan prediksi model dengan nilai aktual pada data uji. *Confusion matrix* membantu untuk memahami bagaimana hasil kinerja dari algoritma yang digunakan [20].

Tabel 1 *Confusion Matrix*

		True Class	
		Positive	***Negative***
Predicted Class	***Positive***	TP	FP
	Negative	FN	TN

Jika data hanya mempunyai 2 kelas , maka akan membentuk tabel *confusion matrix* akan memiliki 2 baris dan 2 kolom. Baris pada tabel 1, merupakan hasil dari klasifikasi algoritma dan

ISSN 1410-7228 (Versi Cetak)
ISSN 2716-4233 (Versi Elektronik)

kolom merupakan hasil sebenarnya. Sebagai contoh, tabel 1 adalah hasil *confusion matrix* dari klasifikasi mahasiswa yang lulus tepat waktu dan tidak lulus tepat waktu. TP (*True Positive*) adalah hasil prediksi algoritma yang memprediksi kelas positif dengan benar. Sebagai contoh TP bernilai banyaknya mahasiswa yang diklasifikasikan lulus tepat waktu, dan kelas sebenarnnya memang lulus tepat waktu. FP (*False Positive*) adalah hasil prediksi algoritma bernilai positif, tetapi kelas sebenarnya adalah negatif. Sebagai contoh FP bernilai banyaknya mahasiswa yang diklasifikasikan lulus tepat waktu, tetapi sebenarnya tidak lulus tepat waktu.

FN (*False Negative*) merupakan hasil prediksi algoritma yang bernilai negatif, tetapi kelas sebenarnya adalah positif. Sebagai contoh FN bernilai banyaknya mahasiswa yang diklasifikasikan tidak lulus tepat waktu, tetapi sebenarnya lulus tepat waktu. TN (*True Negative*) merupakan hasil prediksi yang memprediksi kelas negatif dengan benar. Sebagai contoh TN bernilai banyaknya mahasiswa yang diklasifikasikan tidak lulus tepat waktu, yang memang sebenarnya tidak lulus tepat waktu.

Tabel *confusion matrix* juga dapat menghitung performa sebuah algoritma yaitu akurasi, presisi, *recall*, dan *F1-score*. Persamaan (8) dapat digunakan untuk menghitung nilai akurasi. Nilai akurasi sendiri adalah tingkat proporsi dari total prediksi yang benar dengan nilai aktual. Persamaan (9) adalah rumus yang digunakan untuk menghitung nilai presisi. Presisi sendiri merupakan rasio dari prediksi positif yang benar dibandingkan dengan total prediksi positif. Persamaan (10) adalah rumus untuk mencari nilai *recall*. *Recall* sendiri adalah nilai rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Persamaan (11) adalah rumus yang digunakan untuk mencari nilai *F1-score*. *F1-score* adalah rata-rata dari nilai presisi dan nilai *recall*.

$$Akurasi = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{8}$$

$$Presisi = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

$$F1Score = 2 \times \frac{Presisi \times Recall}{Presisi + Recall} \tag{11}$$

3. HASIL DAN PEMBAHASAN

3.1 *Dataset*

Dataset yang digunakan dalam penelitian ini adalah *dataset Water Quality* yang didapat dari situs Kaggle. Berikut adalah link *dataset* https://www.kaggle.com/*datasets*/adityakadiwal/*water-potability*/data. *Dataset* ini berisi tentang kualitas air dengan format csv. *Dataset* ini memiliki 3276 baris dan 10 kolom, yang terdiri dari 9 fitur dan 1 label. Tipe data untuk masing masing fitur adalah *float*, lalu tipe data label adalah *integer*.

Tabel 2 *Dataset Water Quality*

ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
	204.9	20791.3	7.3	368.5	564.3	10.4	87.0	3.0	0
3.7	129.4	18630.1	6.6		592.9	15.2	56.3	4.5	0
8.0	224.2	19909.5	9.3		418.6	16.9	66.4	3.0	0
8.5	214.4	22018.4	8.1	356.9	363.3	18.4	100.3	4.6	0
8.0	181.1	17978.6	6.5	310.1	398.4	11.8	32.0	4.1	0
9.0	188.3	28748.7	7.5	326.7	280.5	8.4	54.9	2.0	0
10.2	248.1	28749.7	7.5	393.67	283.7	13.8	84.6	2.7	0
8.0	203.4	13672.1	4.6	303.3	474.6	12.4	62.8	4.4	0
	118.9	14285.6	7.8	268.6	389.4	12.7	53.	3.0	0
11.1	227.2	25484.5	9.1	404.0	563.9	17.9	72.0	4.4	0
7.4	165.5	32852.6	7.0	329.6	425.4	15.0	78.7	3.7	0
8.0	218.7	18767.7	8.1		364.1	14.5	76.5	4.0	0
7.1	156.7	18730.8	5.6	282.3	347.7	15.6	79.5	3.4	0
	150.1	27531.4	6.8	299.4	379.8	10.4	78.5	4.6	0

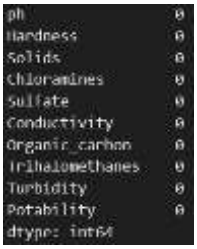
Tabel 2 merupakan isi dari *dataset Water Quality*. Data di tabel 2 sedikit dimodifikasi karena angka di belakang koma terlalu banyak, sehingga dilakukan pembulatan 1 angka dibelakang koma agar dapat ditampilkan. Namun dalam pengolahan datanya, data tidak dibulatkan sama sekali. Fitur pada *dataset* ini adalah *ph, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic_carbon, Trihalomethanes, Turbidity*, dan label dalam *dataset* ini adalah *Potability*.

1. *ph* merupakan derajat keasaman air dari rentang 0 sampai 14.
2. *Hardness* (Kekerasan) merupakan kapasitas air untuk mengendapkan sabun dalam mg/L.
3. *Solids* (Padatan) merupakan jumlah padatan terlarut dalam *ppm*.
4. *Chloramines* (Kloramin) merupakan jumlah kloramin dalam satuan *ppm.*
5. *Sulfate* (Sulfat) merupakan jumlah sulfat yang terlarut dalam mg/L.
6. *Conductivity* (Konduktifitas) merupakan konduktifitas Listrik air dalam μS/cm.
7. *Organic_carbon* (Karbon Organik) merupakan jumlah karbon organik dalam *ppm*.
8. *Trihalomethanes* (Trihalometana) merupakan jumlah Trihalometana dalam μg/L.
9. *Turbidity* (Kekeruhan) merupakan ukuran sifat emisi Cahaya air dalam NTU.\
10. *Potability* (Kelayakan Minum) merupakan suatu ukuran apakah air aman untuk dikonsumsi atau tidak. Nilai -1 menunjukan tidak aman untuk diminum, sedangkan nilai 1 menunjukan bahwa air aman untuk diminum.

Setelah mengunduh data, langkah selanjutnya adalah melakukan pra pemrosesan data. Data secara keseluruhan dilihat apakah memiliki nilai yang hilang menggunakan metode *isnull()*. Hasil metode *isnull* dapat dilihat pada gambar 2.

Gambar 2 Sebelum Mengisi Nilai yang Hilang

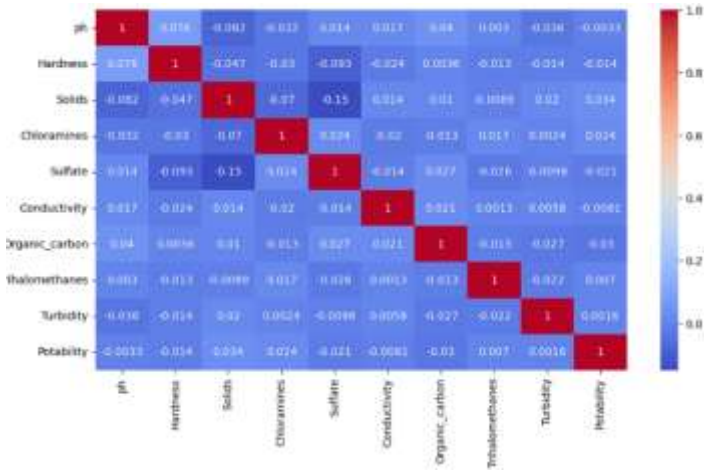
Gambar 2 menunjukan *dataset* memiliki nilai yang hilang yaitu pada kolom *ph, Sulfate*, dan *Trihalomethanes*. *Dataset* diputuskan untuk tidak dihapus nilai yang hilangnya karena jumlah sampel tidak terlalu banyak. Namun Nilai yang hilang ini harus diisi karena algoritma pembelajaran mesin yang digunakan beranggapan bahwa data tidak memiliki nilai yang hilang. Metode yang digunakan untuk mengisi nilai yang hilang pada penelitian ini adalah *fillna()* dengan menggunakan nilai rata-rata.



Gambar 3 Setelah Mengisi Nilai yang Hilang

Dapat dilihat pada gambar 3 bahwa sudah tidak ada lagi nilai yang hilang setelah mengisinya dengan nilai rata-rata dari masing masing kolom. Selanjutnya adalah melihat korelasi dari setiap kolom, apakah ada kolom yang memiliki korelasi yang cukup tinggi. Karena jika ada kolom yang memiliki korelasi yang tinggi, maka ada kemungkinan salah satu kolom akan dihapus. Metode yang digunakan untuk melihat korelasi adalah *sns.heatmap()*.

ISSN 1410-7228 (Versi Cetak)
ISSN 2716-4233 (Versi Elektronik)



Gambar 4 Korelasi Antar Kolom Dengan Heatmap

Dapat dilihat pada gambar 4 bahwa tidak ada kolom yang memiliki korelasi yang tinggi. Nilai korelasi tertinggi yang ada adalah 0.076 yaitu pada kolom *ph* dan *Hardness*. Walaupun demikian, nilai tersebut sangat kecil karna hanya 7.6%. Karena nilai korelasi tertinggi pada kolom hanya 7.6%, maka diputuskan untuk tidak menghapus salah satu dari dua kolom tersebut.

3.2 Analisis Hasil

Tabel yang ditampilkan tidak semuanya karena keterbatas halaman. Metode pencarian grid CV mendapatkan nilai hyperparameter optimal untuk algoritma SVM *linear* sebesar 0.001, menghasilkan akurasi sebesar 61%. Algoritma *decision tree* dengan nilai *max_depth* 6 mencapai akurasi 64%, sedangkan algoritma KNN dengan nilai k 20 mencapai akurasi 59%. Eksperimen dilakukan dengan 16 nilai *random state* yaitu 502,87356, 41427, 63518, 43851, 32848, 7736, 47552, 5279, 40170, 54863, 23909, 11474, 80637, 8395, dan 64822. Skenario eksperimen yang dilakukan meliputi :

a. Komposisi data latih 60% dan data uji 40%.
b. Komposisi data latih 70% dan data uji 30%.
c. Komposisi data latih 80% dan data uji 20%.
d. Komposisi data latih 90% dan data uji 10%.

Tabel 3 Hasil Eksperimen *Decision Tree* dengan Data Latih 60%

Random State	Akurasi	*Precision*	*Recall*	*F1-Score*
502	0,65	0,66	0,56	0,53
87356	0,64	0,61	0,56	0,54
41427	0,65	0,65	0,56	0,52
63518	0,64	0,65	0,57	0,54
43851	0,62	0,6	0,56	0,53
32848	0,64	0,62	0,55	0,51
7736	0,62	0,61	0,53	0,47
47552	0,63	0,62	0,55	0,52
5279	0,64	0,62	0,53	0,46
40170	0,65	0,63	0,54	0,5
54863	0,63	0,59	0,55	0,53
23909	0,63	0,6	0,56	0,55
11474	0,64	0,62	0,57	0,55
80637	0,65	0,7	0,55	0,49
8395	0,64	0,62	0,57	0,55
64822	0,63	0,66	0,56	0,51
Rata-Rata	0,6375	0,62875	0,554375	0,51875

Hasil eksperimen dengan algoritma *decision tree* dengan data latih 60% memiliki hasil paling baik pada nilai rata-rata akurasinya yaitu 0,6375, sedangkan nilai rata-rata paling rendah dalam hasil eksperimen pada tabel 3 adalah nilai rata-rata *F1-Score* dengan nilai 0,51875.

Tabel 4 *Confusion Matrix Decision Tree* dengan Data Latih 60%

		True Class	
		Potable	***Not Potable***
Predicted Class	***Potable***	745	439
	Not Potable	40	87

Tabel 4 merupakan *confusion matrix* untuk random state 64822. Berikut adalah pembuktian perhitungan akurasi menggunakan persamaan (8).

$$Akurasi = \frac{(745 + 87)}{(745 + 87 + 439 + 40)} = 0{,}63$$

Tabel 5 Hasil Eksperimen KNN dengan Data Latih 60%

Random State	Akurasi	Precision	Recall	F1-Score
502	0,6	0,53	0,51	0,46
87356	0,59	0,52	0,51	0,47
41427	0,6	0,52	0,51	0,46
63518	0,6	0,57	0,52	0,45
43851	0,59	0,52	0,5	0,44
32848	0,61	0,54	0,51	0,46
7736	0,58	0,49	0,5	0,45
47552	0,6	0,54	0,51	0,45
5279	0,59	0,5	0,5	0,46
40170	0,59	0,49	0,5	0,46
54863	0,61	0,54	0,52	0,46
23909	0,61	0,55	0,52	0,48
11474	0,61	0,53	0,51	0,46
80637	0,58	0,47	0,49	0,43
8395	0,61	0,54	0,51	0,46
64822	0,59	0,52	0,51	0,44
Rata-Rata	0,5975	0,523125	0,508125	0,455625

Hasil eksperimen dengan algoritma KNN dengan data latih 60% memiliki hasil paling baik pada rata-rata nilai akurasinya yaitu 0,5975, sedangkan nilai paling rendah dalam hasil eksperimen pada tabel 5 adalah rata-rata nilai *F1-Score* dengan nilai 0,455625.

Tabel 6 Confusion Matrix KNN dengan Data Latih 60%

		True Class	
		Potable	**Not Potable**
Predicted Class	**Potable**	717	474
	Not Potable	68	52

Tabel 6 merupakan *confusion matrix* untuk random state 64822. Berikut adalah pembuktian perhitungan akurasi menggunakan persamaan (8).

$$Akurasi = \frac{(717 + 52)}{(717 + 52 + 474 + 68)} = 0{,}589$$

ISSN 1410-7228 (Versi Cetak)
ISSN 2716-4233 (Versi Elektronik)

Tabel 7 Hasil Eksperimen SVM Linear dengan Data Latih 60%

Random State	Akurasi	Precision	Recall	F1-Score
502	0,62	0,81	0,5	0,39
87356	0,61	0,3	0,5	0,38
41427	0,62	0,31	0,5	0,38
63518	0,6	0,3	0,5	0,37
43851	0,61	0,3	0,5	0,38
32848	0,64	0,32	0,5	0,39
7736	0,6	0,3	0,5	0,38
47552	0,62	0,31	0,5	0,38
5279	0,626	0,31	0,5	0,38
40170	0,63	0,32	0,5	0,39
54863	0,61	0,3	0,5	0,38
23909	0,61	0,3	0,5	0,38
11474	0,64	0,32	0,5	0,39
80637	0,61	0,31	0,5	0,38
8395	0,63	0,32	0,5	0,39
64822	0,59	0,3	0,5	0,37
Rata-Rata	0,616625	0,339375	0,5	0,381875

Hasil eksperimen dengan algoritma SVM Linear dengan data latih 60% memiliki hasil paling baik pada nilai rata-rata akurasinya yaitu 0,616625, sedangkan nilai rata-rata paling rendah dalam hasil eksperimen pada tabel 7 adalah nilai rata-rata *precision* dengan nilai 0,339375.

Tabel 8 Hasil Eksperimen Decision Tree dengan Data Latih 70%

Random State	Akurasi	Precision	Recall	F1-Score
502	0,63	0,58	0,53	0,49
87356	0,64	0,64	0,57	0,54
41427	0,66	0,69	0,58	0,55
63518	0,62	0,62	0,55	0,51
43851	0,64	0,61	0,58	0,57
32848	0,63	0,59	0,57	0,57
7736	0,6	0,58	0,58	0,58
47552	0,64	0,62	0,56	0,53
5279	0,64	0,6	0,57	0,56
40170	0,65	0,62	0,56	0,53
54863	0,63	0,62	0,56	0,53
23909	0,61	0,57	0,55	0,53
11474	0,66	0,64	0,56	0,53
80637	0,63	0,65	0,53	0,45
8395	0,64	0,6	0,56	0,55
64822	0,61	0,6	0,55	0,51
Rata-Rata	0,633125	0,614375	0,56	0,533125

Hasil eksperimen dengan algoritma *decision tree* dengan data latih 70% memiliki hasil paling baik pada nilai rata-rata akurasinya yaitu 0,633125, sedangkan nilai rata-rata paling rendah dalam hasil eksperimen pada tabel 8 adalah nilai rata-rata *recall* dengan nilai 0,56.

Tabel 9 Hasil Eksperimen KNN dengan Data Latih 70%

Random State	Akurasi	Precision	Recall	F1-Score
502	0,61	0,54	0,51	0,47
87356	0,58	0,51	0,51	0,47
41427	0,6	0,52	0,51	0,47
63518	0,6	0,56	0,52	0,46
43851	0,59	0,51	0,5	0,44
32848	0,62	0,54	0,52	0,47
7736	0,59	0,53	0,51	0,46
47552	0,61	0,54	0,52	0,47
5279	0,6	0,52	0,51	0,47
40170	0,59	0,5	0,5	0,47
54863	0,59	0,5	0,5	0,44
23909	0,6	0,54	0,52	0,47
11474	0,6	0,49	0,5	0,45
80637	0,6	0,5	0,5	0,44
8395	0,62	0,53	0,52	0,48
64822	0,57	0,5	0,5	0,44
Rata-Rata	0,598125	0,520625	0,509375	0,460625

Hasil eksperimen dengan algoritma KNN dengan data latih 70% memiliki hasil paling baik pada nilai rata-rata akurasinya yaitu 0,598125, sedangkan nilai rata-rata paling rendah dalam hasil eksperimen pada tabel 9 adalah nilai rata-rata *F1-Score* dengan nilai 0,460625.

Tabel 10 Hasil Eksperimen Decision Tree dengan Data Latih 80%

Random State	Akurasi	Precision	Recall	F1-Score
502	0,66	0,65	0,58	0,56
87356	0,64	0,66	0,58	0,55
41427	0,66	0,65	0,57	0,55
63518	0,64	0,6	0,55	0,52
43851	0,66	0,66	0,59	0,57
32848	0,65	0,61	0,56	0,55
7736	0,61	0,58	0,53	0,49
47552	0,63	0,61	0,55	0,51
5279	0,62	0,58	0,54	0,5
40170	0,66	0,64	0,54	0,5
54863	0,61	0,6	0,55	0,53
23909	0,6	0,59	0,53	0,58
11474	0,66	0,61	0,54	0,51
80637	0,63	0,63	0,53	0,46
8395	0,68	0,65	0,58	0,56
64822	0,61	0,63	0,54	0,49
Rata-Rata	0,63875	0,621875	0,55375	0,526875

Hasil eksperimen dengan algoritma *decision tree* dengan data latih 80% memiliki hasil paling baik pada nilai rata-rata akurasinya yaitu 0,63875, sedangkan nilai rata-rata paling rendah dalam hasil eksperimen pada tabel 10 adalah nilai rata-rata *F1-Score* dengan nilai 0,526875.

ISSN 1410-7228 (Versi Cetak)
ISSN 2716-4233 (Versi Elektronik)

Tabel 11 Hasil Eksperimen SVM Linear dengan Data Latih 80%

Random State	Akurasi	Precision	Recall	F1-Score
502	0,62	0,31	0,5	0,38
87356	0,59	0,3	0,5	0,37
41427	0,62	0,31	0,5	0,38
63518	0,63	0,31	0,5	0,39
43851	0,6	0,3	0,5	0,38
32848	0,63	0,32	0,5	0,39
7736	0,61	0,3	0,5	0,38
47552	0,62	0,31	0,5	0,38
5279	0,61	0,31	0,5	0,38
40170	0,64	0,32	0,5	0,39
54863	0,59	0,29	0,5	0,37
23909	0,59	0,29	0,5	0,37
11474	0,65	0,33	0,5	0,39
80637	0,61	0,31	0,5	0,38
8395	0,64	0,32	0,5	0,39
64822	0,59	0,29	0,5	0,37
Rata-Rata	0,615	0,3075	0,5	0,380625

Hasil eksperimen dengan algoritma SVM Linear dengan data latih 80% memiliki hasil paling baik pada nilai rata-rata akurasinya yaitu 0,615, sedangkan nilai rata-rata paling rendah dalam hasil eksperimen pada tabel 11 adalah nilai rata-rata *precision* dengan nilai 0,3075.

Tabel 12 Hasil Eksperimen Decision Tree dengan Data Latih 90%

Random State	Akurasi	Precision	Recall	F1-Score
502	0,65	0,63	0,57	0,56
87356	0,62	0,68	0,55	0,48
41427	0,67	0,69	0,58	0,56
63518	0,67	0,66	0,56	0,52
43851	0,62	0,62	0,55	0,52
32848	0,66	0,68	0,58	0,55
7736	0,62	0,57	0,53	0,48
47552	0,65	0,59	0,53	0,48
5279	0,63	0,67	0,53	0,45
40170	0,64	0,58	0,53	0,48
54863	0,63	0,62	0,55	0,52
23909	0,62	0,64	0,56	0,52
11474	0,67	0,61	0,54	0,51
80637	0,63	0,59	0,57	0,56
8395	0,67	0,62	0,59	0,59
64822	0,66	0,65	0,58	0,56
Rata-Rata	0,644375	0,63125	0,55625	0,52125

Hasil eksperimen dengan algoritma *decision tree* dengan data latih 90% memiliki hasil paling baik pada nilai rata-rata akurasinya yaitu 0,644375, sedangkan nilai rata-rata paling rendah dalam hasil eksperimen pada tabel 12 adalah nilai rata-rata *F1-Score* dengan nilai 0,52125.

Tabel 13 Hasil Eksperimen KNN dengan Data Latih 90%

Random State	Akurasi	Precision	Recall	F1-Score
502	0,59	0,51	0,51	0,45
87356	0,59	0,56	0,52	0,46
41427	0,62	0,56	0,52	0,48
63518	0,61	0,52	0,51	0,48
43851	0,58	0,51	0,5	0,44
32848	0,62	0,58	0,53	0,49
7736	0,62	0,58	0,53	0,48
47552	0,65	0,6	0,54	0,5
5279	0,62	0,57	0,53	0,47
40170	0,6	0,48	0,49	0,44
54863	0,57	0,43	0,48	0,4
23909	0,57	0,51	0,5	0,44
11474	0,6	0,47	0,48	0,45
80637	0,61	0,52	0,51	0,44
8395	0,65	0,57	0,53	0,51
64822	0,58	0,47	0,49	0,43
Rata-Rata	0,605	0,5275	0,510625	0,46

Hasil eksperimen dengan algoritma KNN dengan data latih 90% memiliki hasil paling baik pada nilai rata-rata akurasinya yaitu 0,605, sedangkan nilai rata-rata paling rendah dalam hasil eksperimen pada tabel 13 adalah nilai rata-rata *F1-Score* dengan nilai 0,46.

Tabel 14 Perbandingan Performa Ketiga Algoritma pada Berbagai Komposisi Data

Komposisi data latih	Algoritma	Akurasi	Precision	Recall	F1-Score
60%	Decision Tree	0,6375	0,62875	0,554375	0,51875
	KNN	0,5975	0,523125	0,508125	0,455625
	SVM Linear	0,58625	0,33875	0,5	0,379375
70%	Decision Tree	0,633125	0,614375	0,56	0,533125
	KNN	0,598125	0,520625	0,509375	0,460625
	SVM Linear	0,616625	0,339375	0,5	0,381875
80%	Decision Tree	0,63875	0,621875	0,55375	0,526875
	KNN	0,6	0,526875	0,510625	0,45875
	SVM Linear	0,615	0,3075	0,5	0,380625
90%	Decision Tree	0,644375	0,63125	0,55625	0,52125
	KNN	0,605	0,5275	0,510625	0,46
	SVM Linear	0,619375	0,31	0,5	0,3825

Dari ketiga algoritma yang digunakan pada kasus *dataset Water Quality*, algoritma yang paling baik berdasarkan nilai rata-rata akurasi, *precision*, *recall*, *dan F1-score*, adalah algoritma *decision tree*. Algoritma ini paling baik dengan komposisi data latih 80% dengan nilai rata-rata akurasi 0,63875, *precision* 0,621875, *recall* 0,55375, *F1-Score* 0,526875. Sedangkan algoritma yang paling buruk dalam kasus dalam penelitian ini adalah SVM Linear pada komposisi data latih 60% dengan nilai rata-rata akurasi 0,58625, *precision* 0,33875, *recall* 0,5, *F1-Score* 0,379375.

4. KESIMPULAN

Kesimpulan yang dapat diambil dari eksperimen adalah algoritma yang paling baik dalam menangani kasus klasifikasi *dataset Water Quality* adalah *decision tree* pada komposisi data 80%.

ISSN 1410-7228 (Versi Cetak)
ISSN 2716-4233 (Versi Elektronik)

Sedangkan algoritma paling buruk dalam eksperimen ini adalah SVM Linear pada komposisi data 60%. Saran untuk penelitian selanjutnya adalah menggunakan *dataset* dengan jumlah sampel yang lebih banyak lagi, dan mencoba menggunakan metode lain untuk mencari hasil evaluasi yang lebih tinggi.

DAFTAR PUSTAKA

[1] S. Yudo and N. I. Said, "Kondisi Kualitas Air Sungai Surabaya Studi Kasus: Peningkatan Kualitas Air Baku PDAM Surabaya," *Jurnal Teknologi Lingkungan*, 2019.

[2] A. Fadli, "Analisis Kualitas AIr Bersih di Wilayah Kerja Puskesmas Kepulauan Seribu Utara Berdasarkan Peraturan Menteri Kesehatan Nomor 32 Tahun 2017," *Indonesian Scholar Journal of Nursing and Midwifery Science*, vol. 1 No. 05, pp. 172-180, 2021.

[3] S. Sutisna and M. N. Yuniar, "Klasifikasi Kualitas Air Bersih Menggunakan Metode Naive baiyes," *Jurnal Sains dan Teknologi*, vol. 5 No. 1, pp. 243-246, 2023.

[4] F. N. Sari, R. Razali and I. Ningsih, "ANALISIS KUALITAS AIR SUMUR GALI SEBAGAI SUMBER AIR BERSIH DAN AIR MINUM DI KELURAHAN TIBAN LAMA," *Jurnal Kesehatan Ibnu Sina*, vol. 4 No. 2, pp. 1-7, 2023.

[5] R. N. K. Setioningrum, L. Sulistyorini and W. I. Rahayu, "GAMBARAN KUALITAS AIR BERSIH KAWASAN DOMESTIK DI JAWA TIMUR PADA TAHUN 2019," *Jurnal Ilmu Kesehatan Masyarakat*, vol. 16 Nomor 2, pp. 87-94, 2020.

[6] Y. Agustina and A. Atina, "Analisis Kualitas Air Anak Sungai Sekanak Berdasarkan Parameter Fisika Tahun 2020," *Jurnal Penelitian Fisika dan Terapannya (JUPITER)*, vol. 4 No.1, pp. 13-19, 2022.

[7] S. D. Ramdhani and M. R. T. Laksani, "Analisis Uji Kualitas Air di Sungai Kalidami, Kota Surabaya," *Environmental Pollution Journal*, vol. 4 Nomor 1, 2024.

[8] A. I. Addzikri and F. Rosariawari, "Analisis Kualitas Air Permukaan Sungai Brantas Berdasarkan Parameter Fisik dan Kimia," *Jurnal Sains dan Teknologi*, vol. 2 No. 3, pp. 550-560, 2023.

[9] A. Susanto, M. Zannah, E. K. Putro, A. A. Manuel, W. E. Yochu and R. Mahlisa, "Penilaian Status Kualitas Air Baku untuk Air Minum di Area Concentrating Division PT Freeport Indonesia," *Jurnal Teknologi Lingkungan*, vol. 25 No.1, pp. 88-93, 2024.

[10] M. Faisal and D. M. Atmaja, "KUALITAS AIR PADA SUMBER MATA AIR DI PURA TAMAN DESA SANGGALANGIT SEBAGAI SUMBER AIR MINUM BERBASIS METODE STORET," *Jurnal Pendidikan Geografi Undiksha*, vol. 7 No. 2, pp. 73-84, 2019.

[11] F. O. Awalullaili, D. Ispriyanti and T. Widiharih, "KLASIFIKASI PENYAKIT HIPERTENSI MENGGUNAKAN METODE SVM GRID SEARCH DAN SVM GENETIV ALGORITHM (GA)," *Jurnal Gaussian*, vol. 11 No. 4, pp. 488-498, 2022.

[12] A. Michael, "Komparasi Kombinasi Pre-trained Model dengan SVM pada Klasifikasi Kematangan Kopi Berbasis Citra," *DYNAMIC SAINT*, vol. 7 No 1, pp. 42-48, 2022.

[13] A. N. Utami, "Studi Komparasi Klasifikasi Gagal Ginjal Kronis Menggunakan Algoritma SVM, KNN dan MLP," *Jurnal Ilmiah Matematika*, vol. 8 No. 2, pp. 162-167, 2020

[14] F. T. Admojo, "Klasifikasi Aroma Alkohol Menggunakan Metode KNN," *Indonesian Journal of Data and Science*, vol. 1 No. 2, pp. 34-38, 2020.

[15] A. Ahmad, I. and A. Latief, "Perbandingan Metode KNN Dan LBPH Pada Klasifikasi Daun Herbal," *JURNAL RESTI*, vol. 5 No. 3, pp. 537-564, 2021.

[16] A. D. W. sumari, A. R. Syulistyo and P. I. Mawari , "Klasifikasi Mutu Telur Buurng Puyuh Berdasarkan Warna dan Tekstur Menggunakan Metode K-Nearest Neighbor (KNN) dan Fusi Informasi," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 8 No. 5, pp. 1019-1027, 2021.

[17] R. Indrayani, "ANALISA PERBANDINGAN ALGORITMA NAIVE BAYES DAN DECISION TREE PADA KLASIFIKASI DATA TRANSFUSI DARAH," *Jurnal Ilmiah Teknologi Informasi Terapan*, vol. 5 No. 1, pp. 38-44, 2019.

[18] D. A. Mukhsinin, M. Rafliansyah, S. A. Ibrahim, R. Rahmaddeni and D. Wulandari, "Implementasi Algoritma Decision Tree untuk Rekomendasi FIlm dan Klasifikasi Rating pada Platform Netflix," *Indonesian Journal of Machine Learning and Computer Science*, vol. 4, pp. 570-579, 2024.

[19] K. B. Supriyadi, "Analisis Prediksi Profit Proyek Pekerjaan Plafon Gypsum Menggunakan Klasifikasi Deision Tree," *SYNTAX IDEA*, vol. 4 No. 12, pp. 1715-1730, 2022.

[20] M. A. Shinami and S. Bahri, "Klasifikasi Penyakit Kanker Payudara Menggunakan Metode K-Nearet Neighbors (KNN)," *JURNAL FOURIER*, vol. 12 No.2, pp. 79-85, 2023.